

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**



BÀI TẬP LỚN MÔN JAVA QUẢN LÝ SÁCH



Giảng viên hướng dẫn: Ms. Nguyễn Thị Hồng Mỹ

Sinh viên thực hiện: Trương Văn Thiện

19, tháng 12 năm 2022

Contents

1.	Lời Mở Đầu.....	4
2.	Phân Tích Mô Tả Yêu Cầu	4
2.1	Chức năng Đăng Nhập, Đăng Ký,Đổi Mật Khẩu	4
2.2	Chức năng Menu Người dùng sẽ thực hiện chức năng :.....	4
2.3	Chức năng Quản Lý Sách Người dùng sẽ thực hiện chức năng :	4
2.4	Chức năng Quản Lý Loại Sách	5
3.	Mẫu Thiết Kế Mockup.....	5
3.1	Đăng Nhập	5
3.2	Đăng Ký	5
3.3	Đổi Mật Khẩu	6
3.4	Menu	7
3.5	Quản Lý Sách.....	8
3.6	Quản Lý Loại	8
4.	Sơ Đồ Quan Hệ.....	8
5.	Code Quan Trọng.....	9
5.1	Đăng Nhập	9
5.2	Đăng Ký	9
5.3	Đổi Mật Khẩu	10
5.4	Menu	10
5.5	Quản Lý Sách.....	11
5.6	Quản Lý Loại	13
6.	Hình Ảnh Sản Phẩm.....	14
6.1	Đăng Nhập	14
6.2	Đăng Ký	16
6.3	Đổi Mật Khẩu	17
6.4	Menu	19
6.5	Quản Lý Sách.....	20
6.6	Quản Lý Loại	25

1. Lời Mở Đầu

Qua tìm hiểu thực tế tại một số cửa hàng sách hiện nay em nhận thấy trong các cửa hàng đặc biệt là việc quản lý kho sách đều hoạt động chủ yếu bằng việc ghi chép thủ công thông qua giấy tờ, sổ sách. Do tính đặc thù công việc nên quản lý sách bằng cách như vậy là khá rắc rối, phức tạp và tốn rất nhiều thời gian. Điều đó có thể dẫn đến nhiều sai sót trong việc quản lý và số lượng nhân viên cần cho công việc là tương đối nhiều. Đối với việc quản lý qua sổ sách thì tìm kiếm và thống kê là rất khó khăn, mất thời gian và số lượng thuộc tính được quản lý cũng hạn chế ví dụ như: tác giả, nhà xuất bản, năm xuất bản... Từ đó gây ra khó khăn trong việc tổng kết báo cáo hàng tháng, hàng quý, hàng năm trong quá trình kinh doanh cho ban giám đốc dẫn đến hiệu quả công việc thấp, sai sót trong công việc dẫn đến lợi nhuận mang lại từ cửa hàng là không cao. Cho nên yêu cầu đặt ra là xây dựng một phương thức quản lý mang lại tính nhanh gọn, hiệu quả và quan trọng là chuẩn xác thực sự cần thiết. Dựa trên những kiến thức mà em thu thập được từ các môn học kết hợp các kỹ năng cần thiết, nhóm em xây dựng “Phần mềm quản lý sách”. Hy vọng rằng phần mềm này sẽ giúp giải quyết được các vấn đề tồn tại trong quá trình quản lý kho sách và một số công việc liên quan đến cửa hàng sách. Việc xây dựng thành công phần mềm này là sự tận tình giảng dạy của các giảng viên trong quá trình học tập tại trường đặc biệt là cô hướng dẫn đồ án Hồng Mỹ. Vì thời gian thực hiện đề tài này không dài và kiến thức còn hạn hẹp nên không thể tránh khỏi những thiếu sót, em hy vọng rằng sẽ nhận được sự đóng góp ý kiến của cô góp phần hoàn thiện đề tài này. Em xin chân thành cảm ơn !!!

2. Phân Tích Mô Tả Yêu Cầu

2.1 Chức năng Đăng Nhập, Đăng Ký, Đổi Mật Khẩu

- Người dùng sẽ có thể đăng ký tài khoản mới hoặc đăng nhập tài khoản đã đăng ký từ trước đó.
- Ngoài ra người dùng có thể đổi mật khẩu của 1 tài khoản đã tạo trước đó

2.2 Chức năng Menu

Người dùng sẽ thực hiện chức năng :

- Chuyển sang frame Quản Lý Sách
- Chuyển sang frame Quản Lý Loại Sách
- Đăng Xuất ra khỏi tài khoản

2.3 Chức năng Quản Lý Sách

Người dùng sẽ thực hiện chức năng :

- Thêm, sửa xóa 1 cuốn sách
- Sắp xếp danh sách sách theo tiêu chí
- Tìm kiếm 1 cuốn sách trong danh sách
- Đăng xuất ra khỏi tài khoản

2.4 Chức năng Quản Lý Loại Sách

Người dùng sẽ thực hiện chức năng :

- Thêm, sửa xóa 1 cuốn sách
- Sắp xếp danh sách sách theo tiêu chí
- Tìm kiếm 1 cuốn sách trong danh sách
- Đăng xuất ra khỏi tài khoản

3. Mẫu Thiết Kế Mockup

3.1 Đăng Nhập

The mockup shows a login form titled "Login". It contains two input fields: "User Name" and "Pasword". Below these fields are three buttons: "Đăng Nhập", "Đăng Ký", and "Đổi Mật Khẩu". Labels with arrows point to these elements: "Nhập tên tài khoản" points to the User Name field, "Nhập mật khẩu" points to the Pasword field, and "Các nút chức năng" points to the three buttons.

Chức năng trang đăng nhập :

- Người dùng có thể đăng nhập hệ thống bằng cách nhập tên tài khoản và mật khẩu tài khoản đã tạo trước sau đó ấn nút đăng nhập.
- Khi chưa có tài khoản người dùng có thể ấn nút đăng ký để chuyển đến trang đăng ký.
- Nếu muốn đổi thông tin mật khẩu người dùng ấn nút đổi mật khẩu.

3.2 Đăng Ký

Register

User Name

password

Confirm password

Đăng Ký Trang Chủ

Nhập tên tài khoản muốn tạo

Nhập mật khẩu

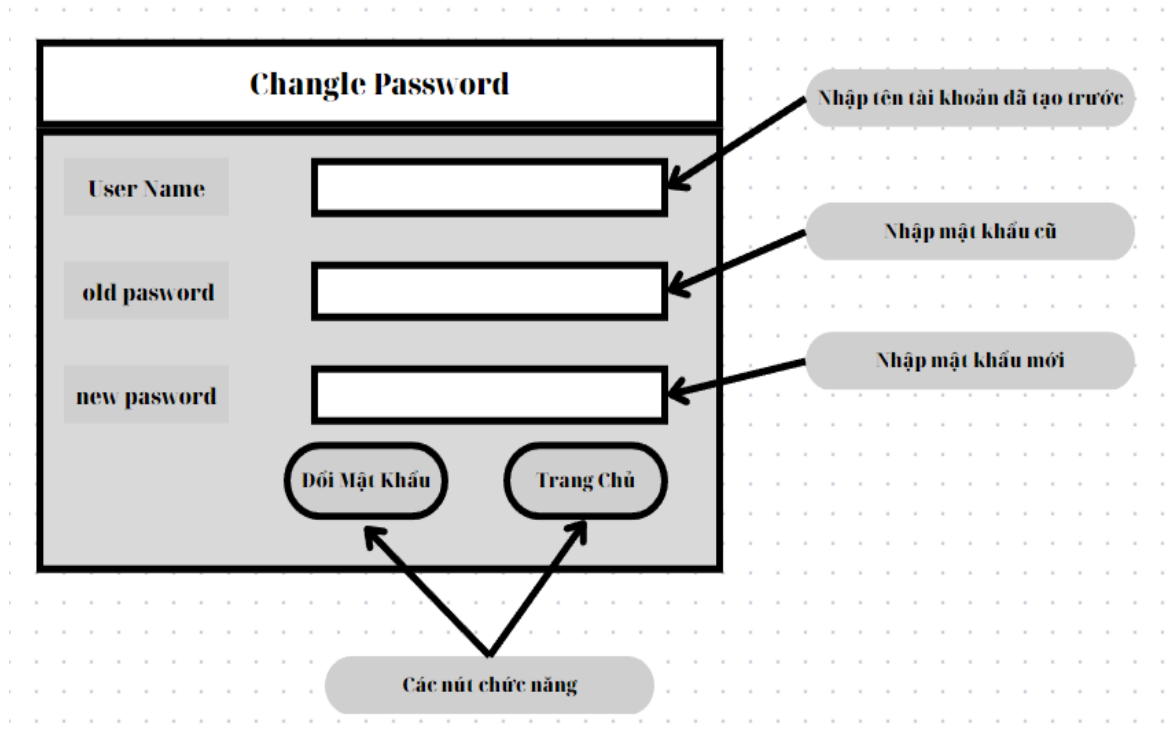
Nhập lại mật khẩu

Các nút chức năng

Chức năng trang đăng ký :

- Ở trang đăng ký người dùng tạo tài khoản bằng cách nhập tên tài khoản của riêng mình sau đó tạo mật khẩu để đăng nhập, để xác nhận mật khẩu cần nhập lại mật khẩu 1 lần nữa. Sau cùng để hoàn tất việc đăng ký người dùng sẽ ấn nút đăng ký.
- Khi đã đăng ký xong hoặc muốn đăng nhập tài khoản khách hàng có thể nhấn nút trang chủ để thoát ra màn hình đăng nhập.

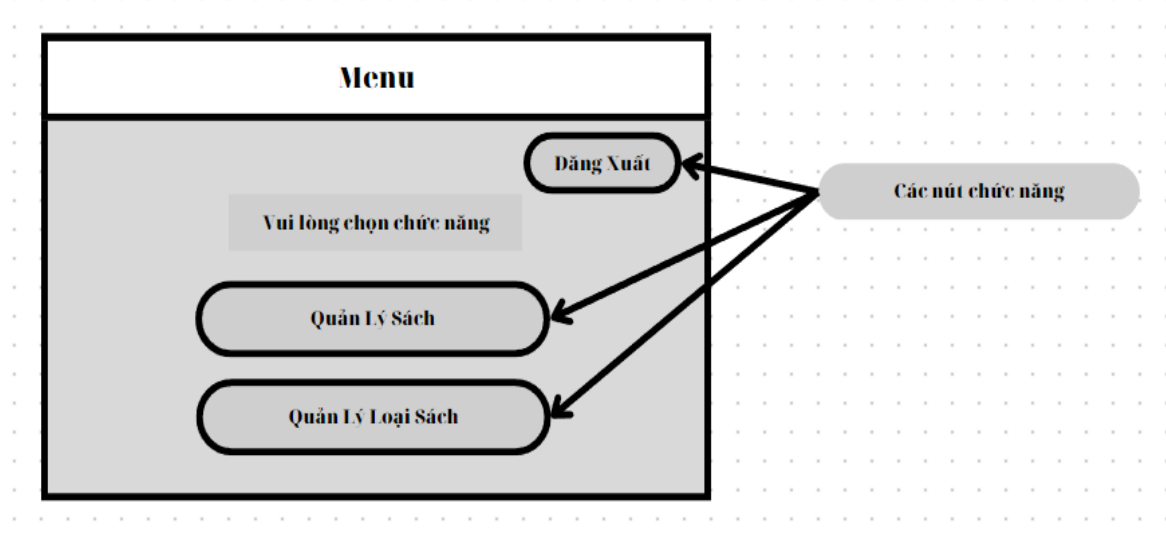
3.3 Đổi Mật Khẩu



Chức năng đổi mật khẩu :

- Ở trang đổi mật khẩu người dùng nhập tên tài khoản đã tạo trước đó sau đó nhập mật khẩu cũ và mật khẩu mới. Sau cùng để hoàn tất việc đổi mật khẩu người dùng sẽ ấn nút đổi mật khẩu.
- Khi đã đổi mật khẩu xong hoặc muốn đăng nhập tài khoản khách hàng có thể nhấn nút trang chủ để thoát ra màn hình đăng nhập.

3.4 Menu



Chức năng Menu :

- Người dùng sẽ ấn các nút chức năng “Quản Lý Sách” hoặc “Quản Lý Loại Sách” để chuyển đến trang chính của các chức năng đó.
- Nút đăng xuất giúp người dùng thoát ra màn hình đăng nhập.

3.5 Quản Lý Sách

THÔNG TIN KHO SÁCH

Đăng Xuất

Mã sách	tên sách	Nhà Xuất bản	giá tiền	mã loại

Mã sách:

tên sách:

Nhà xuất bản:

Giá tiền:

mã loại:

Sắp xếp: tiêu chí sắp xếp
 Tìm:

Thêm Sửa Xóa Mới Thoát

text box nhập và sửa thông tin

Các nút chức năng

thoát ra màn hình đăng nhập

bảng hiện dữ liệu từ database

Sắp xếp thông tin hiện trên bảng

textbox nhập thông tin tìm kiếm

Thoát ra màn hình menu

3.6 Quản Lý Loại

THÔNG TIN LOẠI

Đăng Xuất

Mã sách:

tên sách:

Tìm kiếm:

Thêm Xóa Sửa Mới Thoát

Mã loại	Tên Loại

Sắp xếp: tiêu chí sắp xếp

text box nhập và sửa thông tin

Các nút chức năng

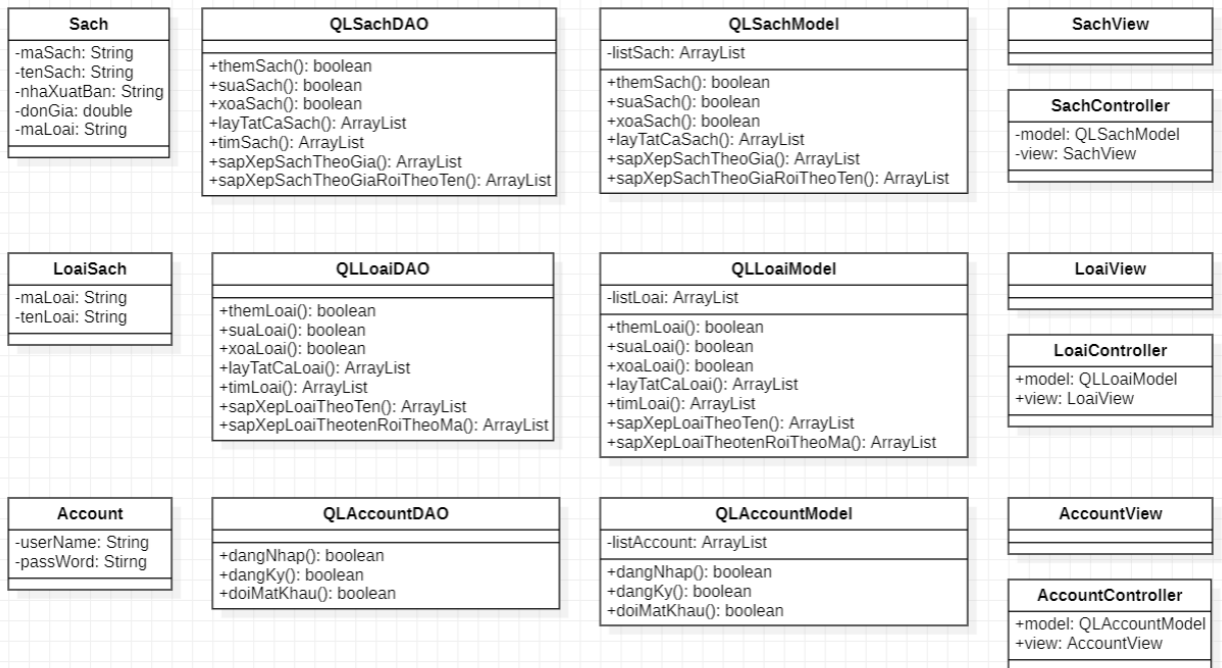
thoát ra màn hình đăng nhập

bảng hiện dữ liệu từ database

Sắp xếp thông tin hiện trên bảng

Thoát ra màn hình menu

4. Sơ Đồ Quan Hệ



5. Code Quan Trọng

5.1 Đăng Nhập

```
private ActionListener dangNhap() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            Account ac = new Account();
            ac.setUsername( username:frameLogin.getTxtUser().getText());
            ac.setPassword( password:String.valueOf( data:frameLogin.getTxtPass().getPassvord()));
            if (frameLogin.getTxtUser().getText().equals( "")) || String.valueOf( data:frameLogin.getTxtPass().getPassvord()).equals( "")) {
                JOptionPane.showMessageDialog( parentComponent: frameLogin, message: "Vui lòng nhập tên tài khoản và mật khẩu");
            } else if (modelLogin.dangNhap( ac) == false) {
                JOptionPane.showMessageDialog( parentComponent: frameLogin, message: "sai tên đăng nhập hoặc mật khẩu");
            } else {
                JOptionPane.showMessageDialog( parentComponent: frameLogin, message: "đăng nhập thành công");
                try {
                    modelLogin.dangNhap( ac);
                    frameLogin.setVisible( false);
                    MenuView view = new MenuView();
                    MenuController control = new MenuController(view);
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog( parentComponent: frameLogin, message: ex.toString());
                }
            }
        }
    };
}
```

5.2 Đăng Ký

```

private ActionListener DangKyListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            Account ac = new Account();
            String user = frameRegister.getTxtUs().getText();
            String Pass1 = String.valueOf( data: frameRegister.getTxtPass1().getPassword());
            String Pass2 = String.valueOf( data: frameRegister.getTxtPass2().getPassword());
            ac.setUserName( username: user);
            ac.setPassword( password: Pass1);
            if (Pass1.equals( anObject: Pass2) == false) {
                JOptionPane.showMessageDialog( parentComponent: frameRegister, message: "Nhập Mật Nhập lại không trùng nhau vui lòng nhập lại!!!");
                frameRegister.getTxtPass2().setText( t: "");
                frameRegister.getTxtPass1().setText( t: "");
            } else if (user.equals( anObject: "") || Pass1.equals( anObject: "") || Pass2.equals( anObject: "")) {
                JOptionPane.showMessageDialog( parentComponent: frameRegister, message: "Nhập thông tin đăng ký!!!");
            }
            else if (modelLogin.dangKy(ac) == false) {
                JOptionPane.showMessageDialog( parentComponent: frameRegister, message: "tên Tài khoản đã được tạo vui lòng nhập tên tài khoản khác!!!");
            }
            else {
                JOptionPane.showMessageDialog( parentComponent: frameRegister, message: "đăng Ký thành công");
                try {
                    modelLogin.dangKy(ac);
                    frameRegister.setVisible( b: false);
                    LoginModel model = new LoginModel();
                    LoginView view = new LoginView();
                    LoginController controller = new LoginController( modelAccount: model, frameAccount: view);
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog( parentComponent: frameRegister, message: ex.toString());
                }
            }
        }
    };
}

```

5.3 Đổi Mật Khẩu

```

private ActionListener doiMatKhau() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String user = frameChange.getTxtUs().getText();
            String oldPass = String.valueOf( data: frameChange.getjPass1().getPassword());
            String newPass = String.valueOf( data: frameChange.getjPass2().getPassword());
            if (user.equals( anObject: "") || oldPass.equals( anObject: "") || newPass.equals( anObject: "")) {
                JOptionPane.showMessageDialog( parentComponent: frameChange, message: "Thông Tin không Được Để Trống");
            } else if (modelChange.doiMatKhau( username: user, oldPass, newPass) == true) {
                try {
                    modelChange.doiMatKhau( username: user, oldPass, newPass);
                    frameChange.getTxtUs().setText( t: "");
                    frameChange.getjPass1().setText( t: "");
                    frameChange.getjPass2().setText( t: "");

                    JOptionPane.showMessageDialog( parentComponent: frameChange, message: "Đổi Thông Tin Thành Công");
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog( parentComponent: frameChange, message: ex.toString());
                }
            } else {
                JOptionPane.showMessageDialog( parentComponent: frameChange, message: "Đổi Thông Tin Không Thành Công");
            }
        }
    };
}

```

5.4 Menu

```

//Chuyển đến chức năng quản lý sách
private ActionListener sachListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            view.setVisible(b: false);
            SachView frameSach = new SachView();
            QLSachModel model = new QLSachModel();
            SachController controller = new SachController(frameSach, modelSach: model);
        }
    };
}

//chuyển đến chức năng quản lý loại
private ActionListener loiListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            view.setVisible(b: false);
            LoiSachView frameLoai = new LoiSachView();
            QLLoiSachModel model = new QLLoiSachModel();
            LoiController controller = new LoiController(view: frameLoai, model);
        }
    };
}

//đăng xuất
private ActionListener logoutListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showConfirmDialog(parentComponent: view, message: "Đăng Xuất???",
            view.setVisible(b: false);
            LoginView frameAcc = new LoginView();
            LoginModel model = new LoginModel();
            LoginController controller = new LoginController(modelAccout: model, frameAccout: frameAcc);
        }
    };
}
}

```

5.5 Quản Lý Sách

```

//hiện bảng thông tin sách
public void showTableSach(ArrayList<Sach> dsSach) {
    DefaultTableModel tableModel = (DefaultTableModel) frameSach.getTblSach().getModel();
    tableModel.setRowCount(rowCount: 0);

    for (Sach sach : dsSach) {
        tableModel.addRow(new Object[]{sach.getMaSach(),
            sach.getTenSach(), sach.getNhaXuatBan(), sach.getDonGia(), sach.getMaLoai()});
    }
    frameSach.getTblSach().setModel(tableModel);
}

//thêm 1 cuốn sách
private ActionListener themListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            Sach sach = new Sach();
            String maSach = frameSach.getTxtMaSach().getText();
            String tenSach = frameSach.getTxtNameSach().getText();
            String nhaXB = frameSach.getTxtNXB().getText();
            double gia = Double.parseDouble(frameSach.getTxtGia().getText());
            String maLoai = frameSach.getTxtMaLoai().getText();
            sach.setMaSach(maSach);
            sach.setTenSach(tenSach);
            sach.setNhaXuatBan(nhaXB);
            sach.setDonGia(gia);
            sach.setMaLoai(maLoai);
            if (modelSach.addSach(sach) == false) {
                JOptionPane.showMessageDialog(parentComponent: frameSach, message: "Không thể thêm mới bạn nhập lại!!!");
            } else if (maSach.equals("") || tenSach.equals("") || nhaXB.equals("") || frameSach.getTxtGia().getText().equals("") ||
                JOptionPane.showMessageDialog(parentComponent: frameSach, message: "Thông tin không được để trống!!!");
            } else {
                try {
                    modelSach.addSach(sach);
                    JOptionPane.showMessageDialog(parentComponent: frameSach, message: "Thêm thông tin thành công");
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(parentComponent: frameSach, message: ex.toString());
                }
            }
            showTableSach(dsSach: modelSach.getAllSach());
        }
    };
}
}

```

```

//sự kiện mouseclick
public MouseAdapter tableListener() {
    return new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            //chon dong
            int index = frameSach.getTblSach().getSelectedRow();
            //lay gia tri cua tung cot moi dong
            String maSach = frameSach.getTblSach().getValueAt(row:index, column:0).toString();
            String tenSach = frameSach.getTblSach().getValueAt(row:index, column:1).toString();
            String nXB = frameSach.getTblSach().getValueAt(row:index, column:2).toString();
            String giaTien = frameSach.getTblSach().getValueAt(row:index, column:3).toString();
            String maLoai = frameSach.getTblSach().getValueAt(row:index, column:4).toString();
            //xuat gia tri ra bang txt
            frameSach.getTxtMaSach().setText(maSach);
            frameSach.getTxtTenSach().setText(tenSach);
            frameSach.getTxtNXB().setText(nXB);
            frameSach.getTxtGia().setText(giaTien);
            frameSach.getTxtMaLoai().setText(maLoai);
            //chuyen txtMaSach qua chi doc
            frameSach.getTxtMaSach().setEnabled(enabled: false);
            //tat them
            frameSach.getBtnAdd().setEnabled(enabled: false);
            //bat xoa sua moi
            frameSach.getBtnDel().setEnabled(enabled: true);
            frameSach.getBtnEdit().setEnabled(enabled: true);
            frameSach.getBtnNew().setEnabled(enabled: true);
        }
    };
}

//Xóa 1 cuốn sách
private ActionListener xoaListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String maSach = frameSach.getTxtMaSach().getText();
            if (!maSach.equals("")) {
                int confirm = JOptionPane.showConfirmDialog(frameSach,
                    "Bạn có muốn xóa Môn Học có mã '" + maSach + "' ?");
                if (confirm == 0) {
                    try {
                        modelSach.xoaSach(maSach);
                        JOptionPane.showMessageDialog(frameSach, message: "Xóa Thành công!");
                    } catch (Exception ex) {
                        JOptionPane.showMessageDialog(frameSach, message: ex.toString());
                    }
                }
            }
        }
    };
}

```

```

//sửa thông tin sách theo mã
private ActionListener editListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            Sach sach = new Sach();
            sach.setMaSach(maSach: frameSach.getTxtMaSach().getText());
            sach.setTenSach(tenSach: frameSach.getTxtTenSach().getText());
            sach.setNhaXuatBan(nhaXuatBan: frameSach.getTxtNXB().getText());
            double gia = Double.parseDouble(gia: frameSach.getTxtGia().getText());
            sach.setDonGia(donGia: gia);
            sach.setMaLoai(maLoai: frameSach.getTxtMaLoai().getText());
            if (modelSach.suaSach(sach) == false) {
                JOptionPane.showMessageDialog(frameSach, message: "Sửa không thành công");
            } else {
                try {
                    modelSach.suaSach(sach);
                    JOptionPane.showMessageDialog(frameSach, message: "Cập nhật thông tin sách thành công");
                    showTableSach(modelSach.getAllSach());
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(frameSach, message: ex.toString());
                }
            }
        }
    };
}

```

```

//tìm kiếm
private ActionListener searchListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String ma = frameSach.getTxtSearch().getText();
            ArrayList<Sach> dsSach = new ArrayList<>();
            dsSach = modelSach.searchByName(ma);
            if (ma == null) {
                JOptionPane.showMessageDialog(parentComponent: frameSach, message: "Nhập tên Sách muốn tìm");
            } else if (dsSach == null) {
                JOptionPane.showMessageDialog(parentComponent: frameSach, message: "Không tìm thấy sách ");
            } else {
                try {
                    modelSach.searchByName(ma);
                    frameSach.getBtnMev().setEnabled(n: true);
                    showTableSach(dsSach);
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(parentComponent: frameSach, message: ex.toString());
                }
            }
        }
    };
}

//sắp xếp
public ItemListener sapXepListener() {
    return new ItemListener() {
        @Override
        public void itemStateChanged(ItemEvent e) {
            int sx = frameSach.getChxSapXep().getSelectedIndex();
            switch (sx) {
                case 1:
                    showTableSach(dsSach: modelSach.sortSachByGia());
                    break;
                case 2:
                    showTableSach(dsSach: modelSach.sortSachByGiaAndTen());
                    break;
                default:
                    showTableSach(dsSach: modelSach.getAllSach());
                    break;
            }
        }
    };
}

//đăng xuất

```

5.6 Quản Lý Loại

```

private ActionListener themListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            LoaiSach loai = new LoaiSach();
            loai.setMaLoai(view.getTxtMaLoai().getText());
            loai.setTenLoai(view.getTxtTenLoai().getText());

            if (model.addLoai(loai) == false) {
                JOptionPane.showMessageDialog(parentComponent: View, message: "Mời nhập lại mã!!!");
            } else if (view.getTxtMaLoai().getText().equals("") || view.getTxtTenLoai().getText().equals("")) {
                JOptionPane.showMessageDialog(parentComponent: View, message: "Thông tin Không được để trống");
            } else {
                try {
                    model.addLoai(loai);
                    JOptionPane.showMessageDialog(parentComponent: View, message: "Thêm thông tin thành công");
                } catch (Exception ex) {
                }
            }
            showTableLoai(dsLoai: model.getAllLoai());
        }
    };
}

//Xóa 1 cuốn sách
private ActionListener xoaListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String maLoai = view.getTxtMaLoai().getText();
            if (!maLoai.equals("")) {
                int confirm = JOptionPane.showConfirmDialog(parentComponent: View,
                    "Bạn có muốn xóa Môn Học có mã " + maLoai + " ?");
                if (confirm == 0) {
                    try {
                        JOptionPane.showMessageDialog(parentComponent: View, "Xóa Thành công loại Có mã : " + maLoai);
                        model.xoaLoai(n: maLoai);
                    } catch (Exception ex) {
                        JOptionPane.showMessageDialog(parentComponent: View, message: ex.toString());
                    }
                }
            }
            showTableLoai(dsLoai: model.getAllLoai());
            view.getTxtMaLoai().setText("");
            view.getTxtTenLoai().setText("");
            view.getTxtMaLoai().setEnabled(enabled: true);
            view.getBtnDelete().setEnabled(n: false);
            view.getBtnEdit().setEnabled(n: false);
        }
    };
}

```

```

//sửa thông tin sách theo mã
private ActionListener editListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            LoaiSach loai = new LoaiSach();
            loai.setMaLoai( view.getTxtMaLoai().getText());
            loai.setTenLoai( view.getTxtTenLoai().getText());
            if (model.guoloi(loai) == false) {
                JOptionPane.showMessageDialog( parentComponent: View, message: "Sửa không thành công");
            } else {
                try {
                    JOptionPane.showMessageDialog( parentComponent: View, message: "Cập nhật thông tin sách thành công");
                    showTableLoai( dsLoai: model.getAllLoai());
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog( parentComponent: View, message: ex.toString());
                }
            }
        }
    };
}

//làm mới JFrame hiển thị
private ActionListener newListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            showTableLoai( dsLoai: model.getAllLoai());
            view.getTxtMaLoai().setText( "");
            view.getTxtTenLoai().setText( "");
            view.getTxtSearch().setText( "");
            view.getBtnAdd().setEnabled( true);
            view.getBtnDelete().setEnabled( false);
            view.getBtnEdit().setEnabled( false);
        }
    };
}

//tìm kiếm
private ActionListener searchListener() {
    return new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            String ma = view.getTxtSearch().getText();
            ArrayList<LoaiSach> dsLoai = new ArrayList<>();
            dsLoai = model.searchByName( new: ma);
            if (ma == null) {
                JOptionPane.showMessageDialog( parentComponent: View, message: "Nhập tên loại muốn tìm");
            } else if (dsLoai.isEmpty()) {
                JOptionPane.showMessageDialog( parentComponent: View, message: "Không tìm thấy loại ");
            } else {
                try {
                    view.getBtnNew().setEnabled( true);
                    showTableLoai(dsLoai);
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog( parentComponent: View, message: ex.toString());
                }
            }
        }
    };
}

//sắp xếp
private ItemListener sapXep() {
    return new ItemListener() {
        @Override
        public void itemStateChanged(ItemEvent e) {
            int sx = view.getChxSapXep().getSelectedIndex();
            switch (sx) {
                case 1:
                    showTableLoai( dsLoai: model.sortLoaiByCode());
                    break;
                case 2:
                    showTableLoai( dsLoai: model.sortLoaiByCodeAndName());
                    break;
                default:
                    showTableLoai( dsLoai: model.getAllLoai());
                    break;
            }
        }
    };
}

```

6. Hình Ảnh Sản Phẩm

6.1 Đăng Nhập

Đăng Nhập

Login

User Name :


PassWord :

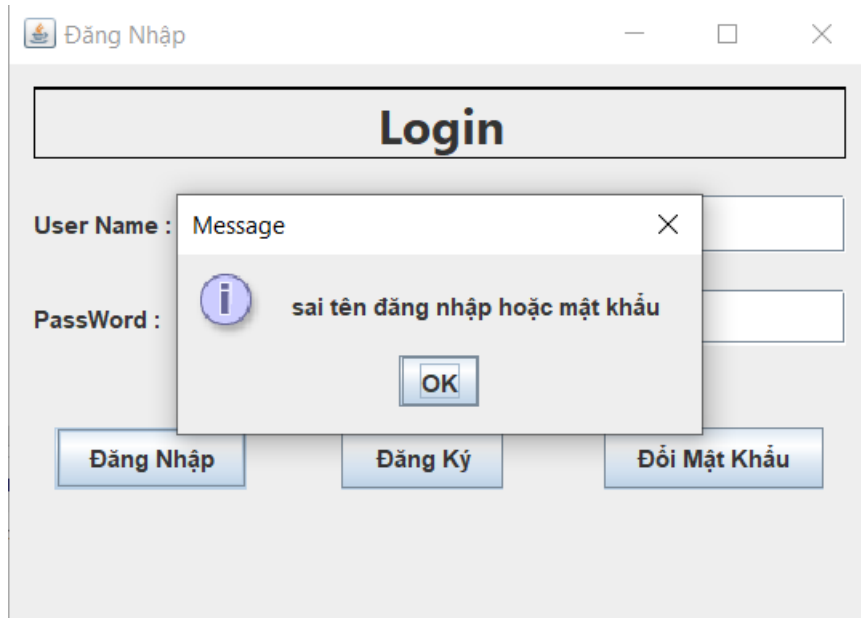
Đăng Nhập

Login

User Name :

PassWord :

 đăng nhập thành công



6.2 Đăng Ký

A screenshot of a web application window titled "Cập Nhật Tài Khoản" (Update Account). The window has a header bar with the title and standard window controls (minimize, maximize, close). Below the header is a section titled "Register" in a large, bold, blue font. Underneath, there are three input fields: "Tên Tài Khoản :" (Account Name) containing the text "admin1", "Mật Khẩu :" (Password) containing three dots "...", and "Nhập Lại Mật Khẩu :" (Repeat Password) containing three dots "...". At the bottom of the register section, there are two buttons: "Đăng Ký" (Register) and "Home".



6.3 Đổi Mật Khẩu

Cập Nhật Tài Khoản

Change Password

Tên Tài Khoản :

Mật Khẩu :

Mật Khẩu Mới :

Cập Nhật Tài Khoản


Change Password

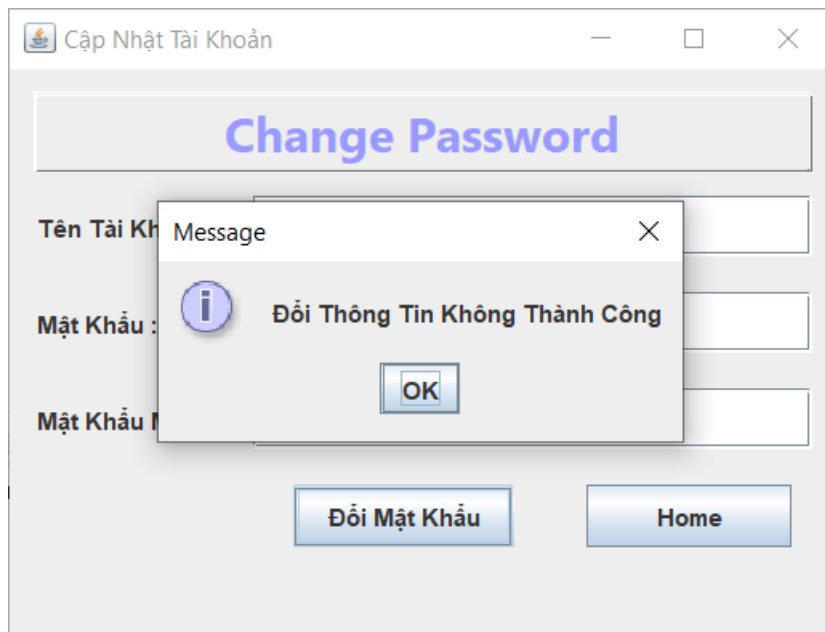
Tên Tài Khoản :

Mật Khẩu :

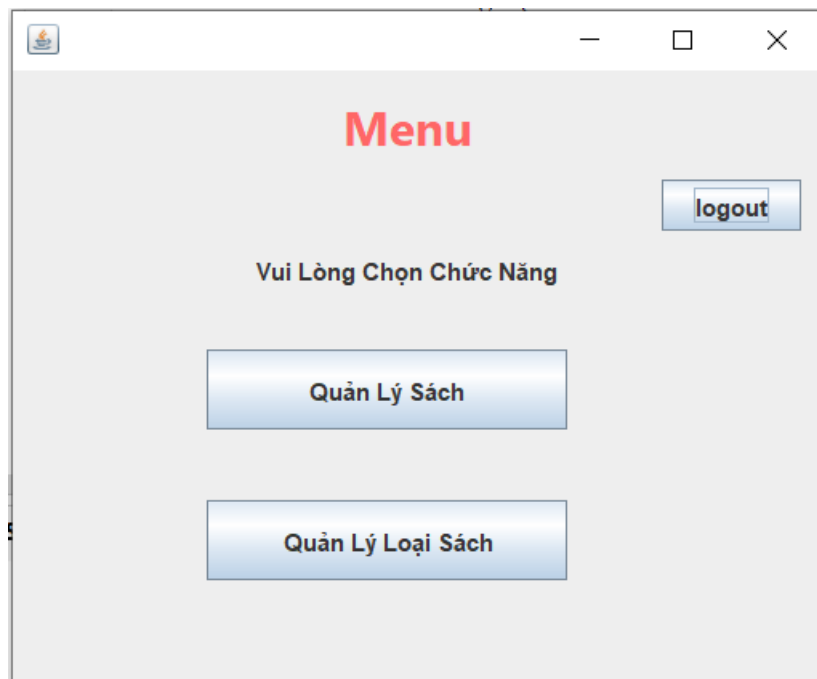
Mật Khẩu Mới :

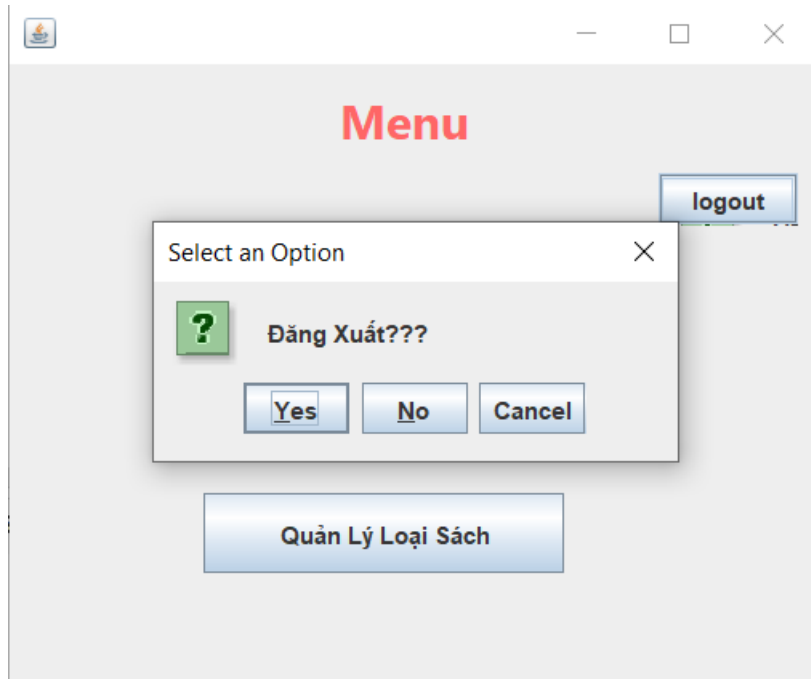
Message

 Thông Tin không Được Để Trống



6.4 Menu





6.5 Quản Lý Sách

The screenshot shows a window titled "THÔNG TIN KHO SÁCH" (Library Information) with a red title bar. The window contains a table of book information, a search section, and a list of actions.

Mã Sách	Tên Sách	Nhà Xuất Bản	Giá Tiền	Mã Loại
S01	lập Trình Java	TDC	50000.0	SGK
S02	Java Nâng Cao	TDC	120000.0	STK
S03	Harry Potter	Kim Đồng	35000.0	SKHVT
S04	Conan	Kim Đồng	15000.0	STT
S05	Nhà Giả Kim	Nhà Xuất Bản Hội Nhà Văn	60000.0	SGK

Below the table, there is a search section with the following fields and buttons:

- Mã Sách :**
- Tên Sách :**
- Nhà Xuất Bản :**
- Đơn Giá :**
- Mã Loại :**
- Sắp Xếp :**
- Nhập thông tin tìm :**
- Thêm** **Sửa** **Mới** **Xóa** **Thoát**
- Tìm Kiếm**

THÔNG TIN KHO SÁCH

Xin Chào :

Logout

Mã Sách	Tên Sách	Nhà Xuất Bản	Giá Tiền	Mã Loại
S01	lập Trình Java	TDC	50000.0	SGK
S02	Java Nâng Cao	TDC	120000.0	STK
S03	Harry Potter	Kim Đồng	25000.0	SKHVT
S04	Conan			STT
S05	Nhà Giả Kim			SGK

Mã Sách :

S05

Tên Sách :

eqwe

Nhà Xuất Bản :

sdad

Đơn Giá :

342

Mã Loại :

d

Chọn Tiêu Chí

Nhập thông tin tìm :

Tìm Kiếm

Thêm

Sửa

Mới

Xóa

Thoát

Message

Mã Sách không được trùng nhau!!!

OK

21 | Page

THÔNG TIN KHO SÁCH

Xin Chào :

Logout

Mã Sách	Tên Sách	Nhà Xuất Bản	Giá Tiền	Mã Loại
S01	lập Trình Java	TDC	50000.0	SGK
S02	Java Nâng Cao	TDC	120000.0	STK
S03	Harry Potter	Kim Đồng	25000.0	SKHVT
S04	Conan			STT
S05	Nhà Giả Kim			SGK

Mã Sách :

S06

Tên Sách :

eqwe

Nhà Xuất Bản :

sdad

Đơn Giá :

342

Mã Loại :

d

Chọn Tiêu Chí

Nhập thông tin tìm :

Tìm Kiếm

Thêm

Sửa

Mới


Xóa

Thoát

Message

Thêm thông tin thành công

OK


— □ ×

THÔNG TIN KHO SÁCH


Xin Chào :

Logout

Mã Sách	Tên Sách	Nhà Xuất Bản	Giá Tiền	Mã Loại
S01	lập Trình Java	TDC	50000.0	SGK
S02	Java Nâng Cao	TDC	120000.0	STK
S03	Harry Potter	Kim Đồng	25000.0	SKHVT
S04	Conan			STT
S05	Nhà Giả Kim			SGK
S06	eqwe			d

Message

×



Cập nhật thông tin sách thành công

OK

Mã Sách :

S06

Chọn Tiêu Chí

▼

Tên Sách :

Kim Loại

Nhập thông tin tìm :

Nhà Xuất Bản :

BBB

Tim Kiếm

Đơn Giá :

342.0

Thêm

Sửa


Mã Loại :

SGK

Mới

Xóa

Thoát



THÔNG TIN KHO SÁCH

Xin Chào :

Logout

Mã Sách	Tên Sách	Nhà Xuất Bản	Giá Tiền	Mã Loại
S01	lập Trình Java	TDC	50000.0	SGK
S02	Java Nâng Cao	TDC	120000.0	STK
S03	Harry Potter	Kim Đồng	25000.0	SKHVT
S04	Conan			STT
S05	Nhà Giả Kim			SGK
S06	Kim Loại			SGK

Mã Sách :

S06

Tên Sách :

Kim Loại

Nhà Xuất Bản :

BBB

Đơn Giá :

342.0

Mã Loại :

SGK

Chọn Tiêu Chí

Thêm

Sửa

Mới

Xóa

Thoát

Nhập thông tin tìm :

Tim Kiếm

Select an Option


?

Bạn có muốn xóa Môn Học có mã 'S06' ?

Yes

No

Cancel



THÔNG TIN KHO SÁCH

Xin Chào :

Logout

Mã Sách	Tên Sách	Nhà Xuất Bản	Giá Tiền	Mã Loại
S01	lập Trình Java	TDC	50000.0	SGK

Mã Sách :

Tên Sách :

Nhà Xuất Bản :

Sắp Xếp :

Chọn Tiêu Chí

Nhập thông tin tìm :

Lập trình

Tim Kiếm

THÔNG TIN KHO SÁCH

Xin Chào : Logout

Mã Sách	Tên Sách	Nhà Xuất Bản	Giá Tiền	Mã Loại
S04	Conan	Kim Đồng	15000.0	STT
S03	Harry Potter	Kim Đồng	35000.0	SKHVT
S01	lập Trình Java	TDC	50000.0	SGK
S05	Nhà Giả Kim	Nhà Xuất Bản Hội Nhà Văn	60000.0	SGK
S02	Java Nâng Cao	TDC	120000.0	STK

Mã Sách :

Tên Sách :

Nhà Xuất Bản :

Đơn Giá :

Mã Loại :

Sắp Xếp : Sắp Xếp Theo Giá

Nhập thông tin tìm :

Tìm Kiếm

Thêm
Sửa

Mới
Xóa
Thoát

6.6 Quản Lý Loại

Thông Kê Loại Sách

Xin Chào : logout

Mã Loại :

Tên Loại :


Tìm

Thêm
Xóa

Sửa
Mới
Thoát

Mã Loại	Tên Loại
SGK	Sách Giáo Khoa
SKHVT	Sách Khoa Học Viễn Tưởng
STK	Sách Tham Khảo
STT	Sách Trình Thâm

Sắp Xếp : Chọn Tiêu Chí


— □ ×

Thống Kê Loại Sách

Xin Chào :

logout

Mã Loại :

ss

Tên Loại :

ss

Tim

Thêm

Xóa

Sửa

Mới

Thoát

Sắp Xếp :


Chọn Tiêu Chí

▼

Tên Loại
Sách Giáo Khoa
Sách Khoa Học Viễn Tưởng
ss
Sách Tham Khảo
Sách Trình Thám


Message

×



Xóa Thành công loại Có mã :ss

OK



— □ ×

Thống Kê Loại Sách

Xin Chào :

logout

Mã Loại :

Tên Loại :

Tim

Tham Khảo

Thêm

Xóa

Sửa

Mới

Thoát

Sắp Xếp :

Chọn Tiêu Chí▼

Mã Loại	Tên Loại
STK	Sách Tham Khảo